DOCUMENT RESUME

ED 282 048 CE 047 205

AUTHOR Hunt, Graham
TITLE Computer-Aided Instruction.
PUB DATE May 87
NOTE 26p.; Paper presented at the Regional Workshop on Technical/Vocational Teacher Training (Chiba City, Japan, May 11-22, 1987).
PUB TYPE Speeches/Conference Papers (150) -- Information Analyses (070)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Authoring Aids (Programing); *Computer Assisted Instruction; *Courseware; Educational Benefits; Evaluation Criteria; Evaluation Methods; Foreign Countries; *Instructional Development; Instructional Effectiveness; Instructional Material Evaluation; Learning Theories; Material Development; Secondary Education; Technical Education; *Vocational Education
IDENTIFIERS *Asia (Southeast); *Instructional Systems Development

ABSTRACT

This report discusses the impact of and presents guidelines for developing a computer-aided instructional (CAI) system. The first section discusses CAI in terms of the need for the countries of Asia to increase their economic self-sufficiency. The second section examines various theories on the nature of learning with special attention to the role of teachers in the learning process. The third section, which covers instructional systems development, is divided into two parts. The first part focuses on models of instructional design underlying CAI (the traditional approach of identifying skills and then subskills and the approach typified by information processing models) and on the importance of mission, accomplishment, performances, and criterion abilities. The second part deals with instructional strategies, with particular emphasis on learner control strategies and the sequence of instructional strategy events (specifying competencies, developing the first epitome, sequencing all remaining epitomes, defining prerequisite abilities, identifying strategy components, planning lesson logic, and authoring the lesson). The fourth section discusses authoring languages and systems, outlines measures of authoring productivity (power, ease of use, output), and presents a case study evaluation of the WISE (WICAT's Interactive System of Education) authoring system. The advantages of CAI (increased control, reduced resource requirements, individualization, timelessness and availability, reduced training time, improved job performance, convenience, the computer as a change agent, increased learning satisfaction, and reduced development time) are explained in the fifth section. (MN)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED282048

APSDEP

MOL

REGIONAL WORKSHOP

ON

TECHNICAL/VOCATIONAL TEACHER TRAINING

(11-22 May 1987 : Chiba City, Japan)

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

COMPUTER-AIDED INSTRUCTION

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

CE 04 7205

Organized by the Asian Development Bank, Manila, Philippines in association with the Asian and Pacific Skill Development Programme (APSDEP) and the Government of Japan

BEST COPY AVAILABLE

Resource Paper

REGIONAL WORKSHOP ON TECHNICAL/VOCATIONAL TEACHER TRAINING

COMPUTER-AIDED INSTRUCTION

GRAHAM HUNT

This paper is circulated as submitted by the author. The views expressed are those of the author and do not reflect the policies of the organizers of the Workshop.

I. INTRODUCTION

1. A challenge of the 80's and 90's for Asia is the need to increase economic self-sufficiency through the development of a technically competent human resource infrastructure. A key to the realization of this goal, as viewed by the 1980 Eighth Conference of Asian and Pacific Labour Ministries in Manila, is the production of technical teachers as "change agents" to meet the manpower requirements of the countries concerned. The optimistic expectation of the Manila Conference unfortunately runs contrary to the realities of many national experiences. The "Club of Rome" report entitled *No Limits to Learning* (Botkin, et al., 1979) in its identification of what it called the "world problematique," found that education and training systems were characterized by an inability to provide individuals with the learning experiences or opportunities which would allow them to take advantage of rapid technological and social change. This kind of condition the report described as "maintenance learning". It characterized this as people learning to adapt or adjust to changes after they had occured. In many cases learning did not occur to any significant degree until some "shock" or catastrophe had taken place. In contrast to this rather inefficient form of learning, the authors identified *innovative* learning as a process involving both anticipatory and participatory learning strategies.

2. Behind the idea of innovative learning is the belief that if learning is to become efficient and successful for the individual, the activities of the learning process and the goals of that learning, must come under the control of the learner. This contrasts with the all too common experience of learners who find themselves struggling in situations over which they have little or no control. Models of instruction for education and training which accept the innovative philosophy for learning ascribe to a belief that learning is not inherently difficult. The complexity of skills which underlie technological competence do not determine whether or not given individuals will be able to master a particular competence. Mastery can be achieved for every individual regardless of the complexity of the knowledge and skills which must be learnt, if they are given control over their learning process and if the outcomes of their learning are clarified and seen to be relevant. The consequences which flow from this perspective are not only likely to significantly influence the array of competencies which might be prerequisite to the requirements for being an accomplished technical teacher, but also the learning strategies they will want to employ in creating trainee competency.

3. The arrival of the silicon chip has heralded far more than just universal access to immensely powerful and increasingly inexpensive computers and computer technology. It has brought about the end of the Industrial Revolution and ushered in the Information Revolution. As Carkhuff (1983) has said, "As simple machinery once enhanced mankind's physical power and freed people from the back-breaking drudgery of physical labour, now the tiny silicon chip promises to enhance mankind's intellectual power and free people from the drudgery of mindnumbing intellectual labour. Like all true revolutions, it introduces the prospect of a new age: the "Age of Productivity".

4. It is within this context of accelerating technological change and complexity that computer-based learning [redacted] has been seen as a panacea. Many believe it points to the start of the second revolution in education. The first commenced 523 years ago with the production of the first printed book (John Gutenberg's Latin Bible in 1454). But this revolutionary cusp in the transfer of knowledge from master to noviciate took a further 360 years and the invention of Frederick Konig's steam driven cylindrical press, to have a widespread impact on the lives of the masses. The second revolution had its genesis in 1944 with Dr Howard Aiken's development of the first general purpose digit computer. But unlike John Gutenburg's first book, the invention of the computer did not provide the solution, only a means to that end. Just as the press needed the skills of literacy to maximise its impact, so the computer will need the the skills of instructional design and courseware development if it is to realize its potential for enabling individuals to chose and control purposeful learning.

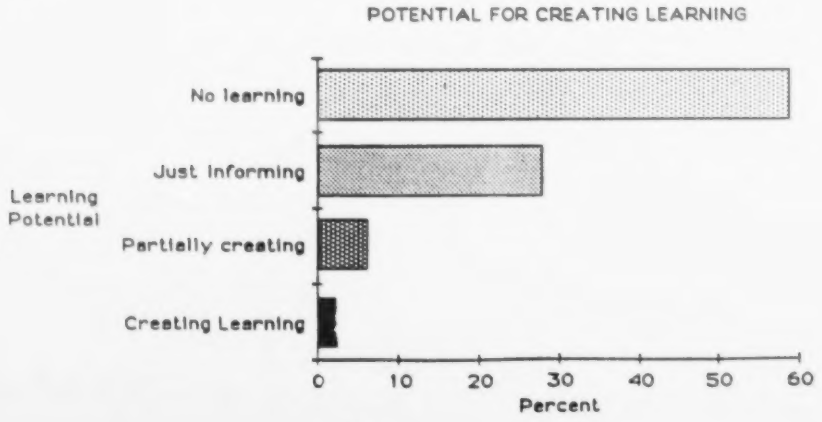
BEST COPY AVAILABLE

II. IN SEARCH OF EFFECTIVE LEARNING

5. Traditionally, two points of view have competed in explaining the nature of learning and teaching. The first suggests that learning is directly attributable to the efficacy of the skills employed by the teacher in the teacher-student learning process. Almost all students, so these protagonists declare, will be able to "master" what they are being taught if they are given adequate instruction (Bloom, 1977). The second view takes the contrary stance and concludes that teaching is a relatively unimportant and vastly overvalued activity (Rogers, 1968). For this school of thought the importance of learning does not lie in the knowledge which can be acquired - knowledge doesn't keep any better than fish! (Whitehead, 1932), but in the process of learning how to learn.

6. A convergence of these views is achieved when the focus of instructional activity is shifted from an exposition of content, or an hedonistic-like pursuit of self-interest, to that of the creation of purposeful learning. In this perspective two determinants for learning predominate; defining what learning is relevant; and prescribing the conditions under which it is most likely to occur. In a study for the Asian Development Bank (Hunt, 1984) this perspective was applied to the identification of attributes which contributed to effective technical teacher instructional performance. Five factors were found to influence how effective teachers were with their students. These were, their belief system or self-concept; their perception of the mission they had in teaching; the range of teaching methods they were prepared to use in the learning process; the degree of student participation they could engender; and of greatest importance, their potential for creating learning (figure 1). This last attribute provided the most significant predictor of teaching competency. Only a tiny percentage of the total sample of instructors demonstrated that they were creating learning. The majority spent most of their time in just giving information or providing a variety of administrative and custodial roles, but not ensuring that substantive learning was taking place. Although these results came from a single study, further evidence (Hunt, 1986) suggests that similar results could be obtained from any country and at every level of the education sytem. The only factor which is likely to substantially modify the impact that teachers have on student learning is the degree to which student's themselves are able to become independent of their teachers for learning. This factor reflects as much upon student "ability" to learn (including their previous success in learning and strategies developed to provide for intrinsic reinforcement), as it does on their access to resources for independent learning. To varying degree, these students can learn inspite of their teachers. But for other students in which these learner characteristics are not so well developed, nor instructional resources such as libraries, self-instructional programmes and audio-visual aids so adequately available, the teacher makes the difference.

Figure 1



7. Two further observations were pertinent from this study. First, more often than not staff are recruited into technical teaching primarily on the basis of their technical expertise, not on whatever attributes they may or may not have for delivering instruction. Secondly, when the question is asked, "How can we improve the quality of learning?" the usual answer given is to "make better teachers". But this answer invariably leads to the wrong solution. The traditional view on what should be done to upgrade the teaching quality of an institution is to increase their academic qualifications and levels of technical expertise. But as this study showed (Table 1), academic and technical qualifications alone have little effect on a teacher's ability to create learning - any more than they might be expected to have in creating computerprogramming competencies in the absence of learning anything about computer programming! Instead, the acquisition of this accomplishment appears to be dependent on two critical facilities. First, the teacher's ability to create specific strategies along the lines of those described by Ashcroft (1983) in his "Determinants of Learning", and secondly, his or her management of a delivery system designed to enable students to control their own learning achievement. On this latter point it is salient to note that inspite all the efforts that have gone into teacher training around the world, the vast majority of teachers still function as the primary source of instruction. Their roles remain analogus to that of doctors of yore. These deliverers of medicine not only diagnosed patients' complaints, but also manufactured the potions to cure the problem. If the patient lived the doctor was acclaimed, and if the patient died, exoneration was accepted on the grounds that it wasn't really his fault. The spirits had determined the fate of the victim and as such had put the patient beyond human intervention. As teachers we hold a more sophisticated view of life. But in the same way, with little outside assistance, we diagnose the learning needs of our students and create and implement the potions (lessons) to meet those needs. If students learn we commend ourselves, and if they don't, we fatalistically accept the result as an indication that their ability and motivation for learning was not up to the task. They just weren't good enough. Their failure is seen as a reflection on their own inadequacies for learning. Never could it be construed that their inability was due to our failure to deliver content that could be processed in such a way that would lead each to succeed. But are we right? Is there an approach to instructional delivery which can facilitate the role of teachers creating learning? We believe there is.

Table 1

Potential for Creating Learning

	Factors	Chi- square	df	Significance
1.	Academic Seniority	15.05	09	Not significant
2.	Sex	09.19	03	Not significant
3.	Age	10.77	15	Not significant
4.	Ethnicity	07.53	09	Not significant
5.	Currently in teacher training	09.17	03	Not significant
6	Academic qualifications	10.74	15	Not significant
7.	Technical qualifications	02.52	03	Not significant

Note:

Significance level = $p<.05$

III. INSTRUCTIONAL SYSTEMS DEVELOPMENT
(A) INSTRUCTIONAL DESIGN

8. A critical need in the search for improved systems for delivering instruction, particularly for CBL, is to develop procedures for defining content and specifying the conditions for translating that content into instructionally potent prescriptions for learning. The evidence to date is that most instructional strategies which underlie CBL systems for developing teaching content (authoring systems), while being mechanically efficient and intellectually seductive, they are instructionally irrelevant (O'Neal(a)). The traditional approach to analyzing complex learning tasks for instructional purposes has been to identify the specific skills and then the sub-skills. This method has sometimes been referred to as a "bottom-up" process. Although the bottom-up and its alternative, "top-down," are both hierarchical in nature, the philosophy and methodology are quite different. In the bottom-up approach, the underlying assumption is that "information flows" work from the environment to the individual. For instance, in analyzing an aircraft pilot's recovery from a stall, the information flow may reveal the manipulation of sub-tasks such as; "push control column forward"; "apply power"; "apply rudder to combat yaw". Each of these elements are identified as reactions of the pilot to what is happening to the aircraft and to systems within the aircraft. They are his or her reactions to an information flow from the environment. In this perspective, the individual is seen to be reactive rather than proactive to the environment. The consequence for learning which flow from this stance is that trainees may acquire sets of skills without any understanding of what those skills mean. Furthermore, in the process of developing curriculum for training, the *abilities* which underlie the specific tasks to be taught and therefore the focus of that instruction, may be hidden or lost in the plethera of individual tasks.

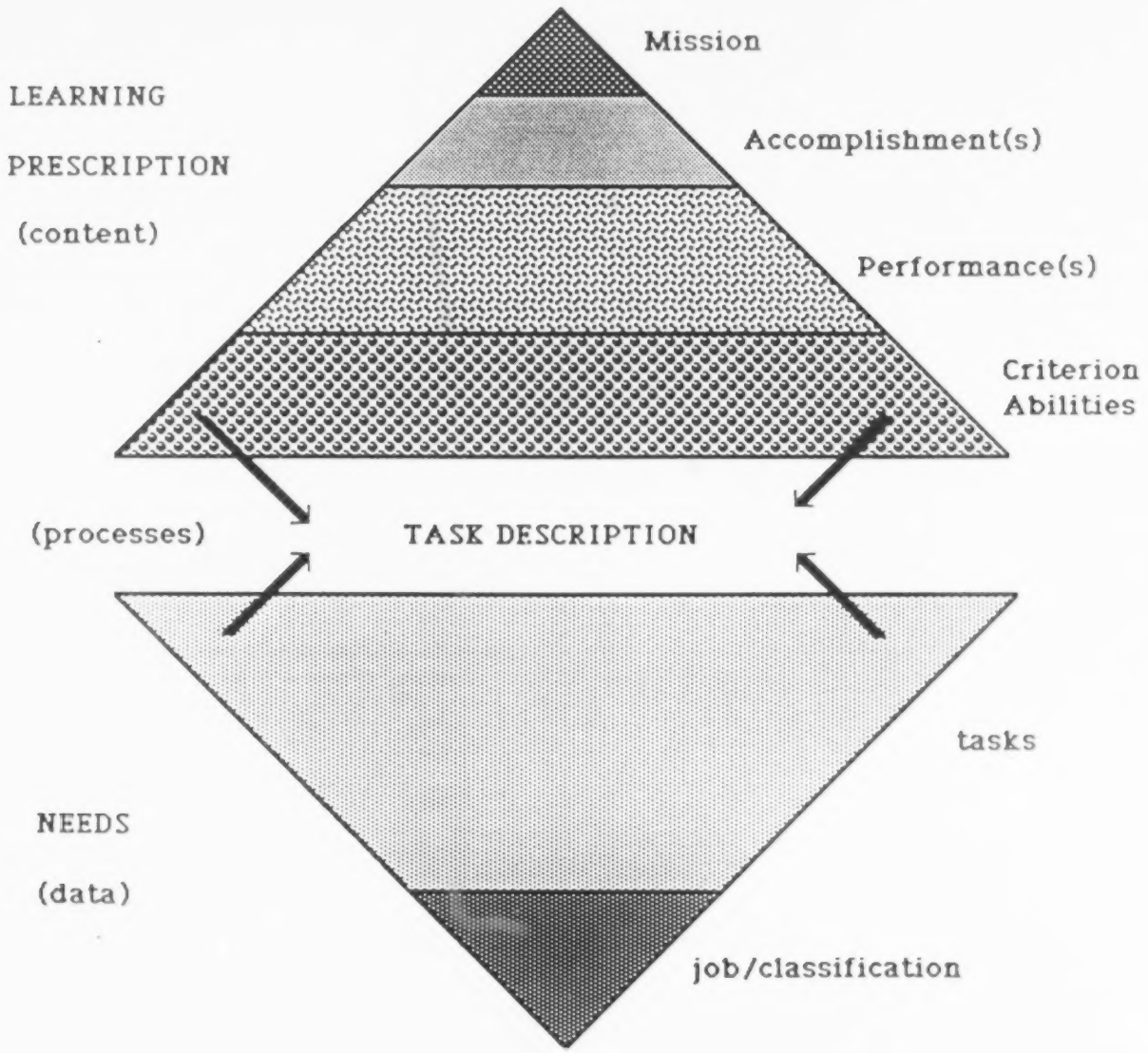
9. The second approach, typified by information-processing models of instructional design such as NEBEAT - Needs Based Education and Training (Hunt, 1984b), define the nature of human competency as a synthesis of processes and products. In models like NEBEAT the primacy of tasks is reduced to a point at which their relevance is limited to defining the contexts within which the abilities (criterion abilities) underlying a particular competency will be demonstrated. Competency descriptions (which share some similarities with Mager styled Instructional Objectives (Mager, 1962)) identify both the abilities which provide the focus for instructional development and the task-related contexts which provide the parameters for those abilities. The pyramid mirror image (figure 2) of the NEBEAT model reflects the interactive nature of the abilities to be learnt and the contexts in which they will be displayed. The identification of individual content factors in the vertical pyramids learning prescription is the tactical result of a needs assessment process. These factors are represented as a series of ever widening and increasingly inclusive statements of human potential.

10. ***Mission.*** At the apex of the pyramid is the mission or overriding goal of the organization's (institution) activity. This is the purpose to which all the accumulating activities are directed. These mission statements claim their validity in the degree to which all participants, from policy makers to job incumbents - teachers, etc., can agree on their usefulness in providing direction and purpose. In vocational teacher training such a mission might be:

> *"To inculcate professional and managerial skills to vocational educatior lecturers so that they can develop competencies within their students which will enhance their employment potential and contribute to the socia and economic well being of the country"*

Figure 2

NEBEAT Instructional Prescription Model



11. **Accomplishment**. This next level of the hierarchy provides statements which are descriptive of an amalgam of individual performances which contribute to the "accomplished" execution of a range of related content abilities. In some ways these accomplishment statements are analogous to "job" statements in a job analysis. However, unlike job analysis, the focus in this kind of competency hierarchy is on the psychological processes which underlie the content of the complex performances. The achievement of a teacher training mission may involve the development of at least five accomplishments; "being a trainer"; "being a learning facilitator"; "being a content expert"; "being a learning resource manager"; and "being a counsellor/helper". Each of these accomplishments are presented as "generic" statements. To be made explicit in a task description they would need to be further defined by their constituent performances and qualified by the contextual parameters (equipment, procedures, conditions and standards) as dictated by any given individual task.

12. **Performances**. These are the specific content areas subsumed within each accomplishment. They are conceptually similar in type to accomplishments, except that they are far more restricted in their domain of achievement. As such they provide a convenient focus for specifying competencies in the instructional design phase of courseware development. An accomplishment like "being a learning facilitator" may have have three performances - "classroom management strategies" (using psychologically-based principles for establishing, maintaining and changing particular kinds of behaviour); "creating learning" (by incorporating three student-centred cycles for instruction into the delivery process); and "accommodating learner characteristics" (identifying pertinent learner attributes such as level of motivation and using this information to develop particular instructional strategies which enhance learning achievement for that type of characteristic). As with accomplishments, each performance is presented as a generic content domain requiring further specification of the competency being identified. This is achieved by defining the specific criterion abilities which make up the performance.

13. **Criterion Abilities**. These are the *essential* abilities which underlie a performance. As prescriptive content variables they reflect the particular nature of the capacity which will need to be learnt by the individual if he or she is going to be able to carry out a particular task. This is different to the action statement found in most traditionally written instructional objectives. They simply present the specific skills to be learnt without identifying the underlying ability which is prerequisite to the skill being demonstrated (Monjan and Grassner, 1979). It is the reluctance of this kind of objective to allow for the identification of learning processes (the criterion abilities) in preference to simply stating the products (the task-related knowledge and skills) that has led to so many of these objectives stating quite trivial behaviour (Hunt, 1986).

14. Criterion abilities have these instructional characteristics when included in a competency description of an instructional design prescription: (1) They are the essential factors or processes invloved in the performance; (2) they constitute the ability needed to engage in the process which makes it possible to master the performance and become accomplished in a range of competencies; (3) they are generic - that is, they can be applied to a variety of performances and contexts; (4) they are identified independently of any task description; and (5) they are the final, ultimate and delimiting (hence the word criterion) abilities in the performance.

15. Criterion abilities may be unique in relation to a performance or highly pervaisive. For instance in pilot training"aeronautical decision-making" is an ability which underlies almost every performance in flying. As a pervaisive ability in a competency specification (see figure 3) it receives a *low* general weighting. However, in a specific task such as "complete a flight plan" it may be moderately important to the ability of mastering "flight planning" and as such could be indicated by a *medium* specific weighting. On the other hand a relatively unique ability (in relation to the sum of performances being taught), in this example, "calculations", could have its uniqueness indicated by assigning it with a *high* relative weighting for the entire course (general), as well as a *high* relative weighting for the specific performance.

Figure 3

COMPETENCY SPECIFICATION

PERFORMANCE

INSTRUCTIONAL RELEVANCE

CRITERION ABILITIES	RELATIVE WEIGHTINGS (GENERAL)	(SPECIFIC)
Stating Objectives	Low	High
Discussing Objectives	Low	Medium
Providing "focusing" problems	High	High
Students "internalizing" Objectives	Medium	High

TASK DESCRIPTION

"Using micro teaching classes, technical teacher trainees will ensurethat objectives for the lesson they are about to teach are understood. The teacher trainee's performance will be judged to be acceptable when he or she has each student in the class participating in such a manner that their behaviour demonstrates they understand and appreciate what it is they are expected to learn."

16. Any number of competency specifications can be generated from an instructional prescription. The learning prescription component (pyramid) defining the content abilities is a "constant" in the prescriptive process. That is, once these factors have been identified from a mission's accomplishment, they remain basically unchanged, regardless of the range of applications (tasks) that might be developed from the prescription. Eventually artifically intelligent tools (systems that can understand natural colloquial languages in contrast to computer languages and can learn to make judgements based on prior experience) will be able to collect and translate data from needs assessment surveys into descriptions of criterion abilities and thereby reduce the inherent subjectivity of the process. The inverted pyramid, or data component of the prescription provides the "variable" input. Each time a piece of equipment, operating procedure, or standard is changed, that information is defined and described in the "task description". Thus, the task description component of the competency specification may change from lesson to lesson, and from one courseware package to the next. But the learning prescription will remain unaltered until the abilities themselves are changed.

III. INSTRUCTIONAL SYSTEMS DEVELOPMENT
(B) INSTRUCTIONAL STRATEGIES

17. One of the major outcomes of instructional systems development (ISD) has been the emergence of a research-based set of instructional strategies. Even as recently as the 70's it was not unusual to find course developers faced with teaching 1000 instructional objectives and writing 1000 individual lessons for those objectives. The degree of redundancy in training was immense! As O'Neal (undated) has observed, in his experience of developing instruction and training for over 10,000 objectives in education and training, between 85% and 90% of them could be successfully taught from a library of less than 12 strategies. In the experience of these authors, a one-year full-time course in electronics-maintenance services with over 480 individual objectives, using the NEBEAT ISD model, resulted in 3 accomplishments and 12 performances (Hunt, 1986). Had these 480 odd objectives been subjected to traditional syllabus development, there is no knowing how many of them would really have been mastered and with what confidence tradespeople could have been said to be "accomplished" in the context of their jobs.

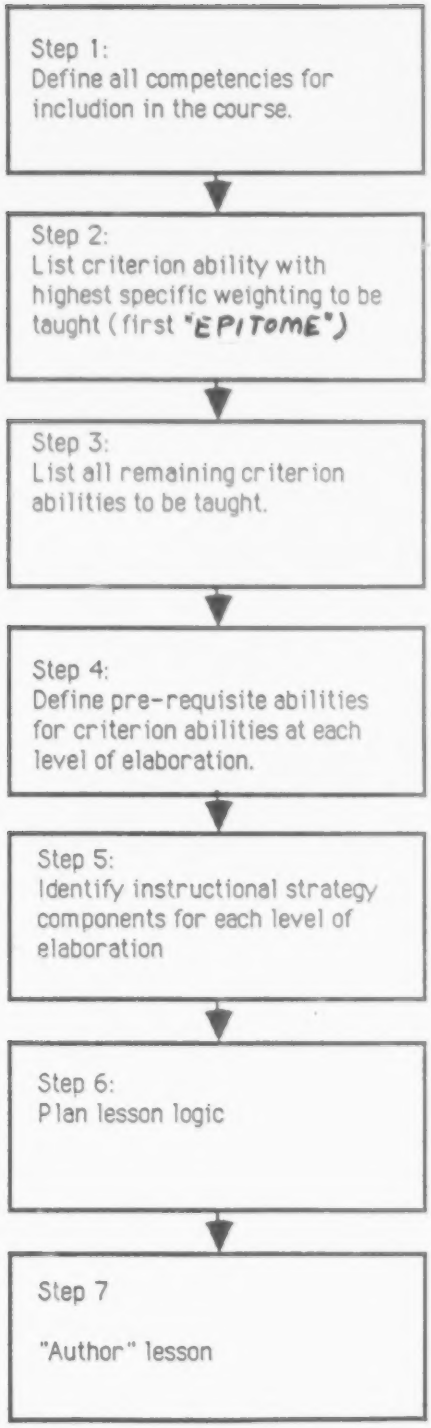
18. There are a number of instructional strategy models which have been applied to CBL courseware development. They include mastery, adaptive, learner control, and aptitude x treatment interactions. The mastery model (Block, 1971) provides for pre and posttests to demonstrate whether the learner has achieved the desired mastery level. If he or she hasn't, practice is provided until that level of performance has been attained. The adaptive model is an extension of the "branching" programme of teaching machines days (Lumsdaine and Glaser, 1960; Crowder, 1962). Here the student is branched into "frames" of instruction based on various levels of difficulty until a level is reached at which the student can proceed without incurring a significant amount of error. In the learner control model students are given the responsibility for managing their own learning sequencing and choosing which strategy components they wish to use in studying, when and for how long (Reigeluth, 1979; Bunderson, 1980). In the aptitude x treatment interaction (ATI) model (Cronbach and Snow, 1977), methods of instruction (treatments) are prescribed on the basis of student characteristics (aptitudes) on the assumption that the effectiveness of instructional methods depend on the interaction with student characteristics. Each of these models is representative of a range of theories on instructional design. One which has provided a lot of promise for the development of computer-based instruction is Reigeluth's learner control or "elaboration" theory. A modified version of his model is described more fully here because of its attractiveness in linking the instructional design phase of ISD to the computer-based authoring component of instructional delivery.

Learner Control Strategies

19. Reigeluth frequently presents his model by describing the analogy of looking at instruction through a zoom lens. It begins with a wide-angle view which shows only the major parts of the picture and their interrelationships. There is little detail. The complexity and detail can be gradually increased by zooming into the parts of interest. As an instructional strategy the zooming process is made up of a number of discrete levels of observation. It starts by taking a wide-angle "organizing" view of the content (ability) to be studied. This is called an *epitome*. As the content is *elaborated* in a simple-to-complex sequence of teaching steps the ideas are presented as concepts, principles or procedures at a concrete, meaningful, application level of utility. We have modified his model somewhat in order to effectively provide a link between the process of competency specification and the strategies of authoring. The sequence of instructional strategy events is illustrated in figure 4.

20. **Step 1 Competency Specification.** The components in this step have been described previously. It should be noted that the criterion abilities form the key focus. The specific procedures, conditions and standards for each ability are developed at Step 5.

Figure 4


Step 1:
Define all competencies for includion in the course.
Step 2:
List criterion ability with highest specific weighting to be taught (first "EPITOME")
Step 3:
List all remaining criterion abilities to be taught.
Step 4:
Define pre-requisite abilities for criterion abilities at each level of elaboration.
Step 5:
Identify instructional strategy components for each level of elaboration
Step 6:
Plan lesson logic
Step 7
"Author" lesson

21. **Step 2 Develop first Epitome.** Locate the criterion competency with the highest general weighting - its measure of uniqueness in relation to the abilities underlying all performances (it may not have the highest specific weighting). Check to see in which other

performances it occurs. This is the first thing to be taught and should be sufficiently inclusive as to be able to relate to the learner's previous knowledge and experience and be capable of translating to new knowledge and experiences.

22. **Step 3 Sequence all remaining** Epitomes. Work from criterion abilities with high general weightings to those with low general weightings, and those with high specific weightings to those with low specific weightings. An analysis of these abilities will provide indicative information on their relative contribution to the abilities which underlie given performances. Prioritize each in terms of the extent to which they contribute to the attainment of performances and mastery of the overall accomplishment.

23. **Step 4 Define pre-requisite abilities**. This step will expand the epitome by including all the sub abilities for an epitome at the first level, and then all the sub-sub abilities for the expanded epitome of level 1 at level 2, etc.

24. **Step 5 Identify strategy components.** These will include the use of seven major strategy components: (1) an elaborative sequence of simple-to-complex idea building, relating to the particular nature of the course; (2) learning pre-requisite sequences in relation to given criterion abilties; (3) summarizers; (4) synthesizers; (5) analogies; (6) cognitive strategy activators; (7) and a learner control format which enables the individual to select and sequence the content of the lesson to be learnt and the pace and practice to be used.

25. **Step 6 Plan lesson Logic**. This stage defines the kinds of frames and the lesson logic which will analyze student responses, provide feedback for the student, initiate audio and video sequences and branch the student to other parts of the course.

26. **Step 7 Author the lesson**. Authoring has particular connotations for CBL. It usually means the process of programming, debugging, and testing of courseware. As a major component of any CBL delivery strategy, a detailed examination of authoring will follow shortly.

27. The key feature for instructional sequencing in learner centered models is the way in which the epitome (or lesson which epitomizes a criterion ability) is presented and the learner initiated responses which can be made. Like Ashcroft's (1983) three learner centered factors: goals, involvement and effects, there are a number of components which can be built into the presentation of an epitome. Reigeluth (1983) has suggested that an epitome may start with a motivational-strategy component such as the creation of an incongruity. This might be followed by an analogy and then by an examination of all the learning prerequisites which will need to be mastered by the target learner population. Individual learner aptitude features, such as high ratios of feedback for high anxious students (say in the teaching of statistics in a test and measurement course for technical teachers) may need to be made available for the learner to control. These suggestions follow the hypothesis that Merrill (1980) and others have postulated for a metacognition model inside each learner that orchestrates how the learner chooses to study to learn.

28. The development of effective instructional strategy procedures which provide prescriptive information on how to match the information-processing styles of learners to the design specifications of the to-be-acquired performance abilities in ways which provide the instructional designer with the tactical blueprints for authoring is still in its infancy. However, models such as those described above are indicative of significant advances in the science of instruction. And as O'Neal (undated) has said, "If a step-by-step procedural job aid supported by individualized instruction and helps can be created to aid in the correct application of of the instructional strategy by CBT developers on real world development projects, then clearly this strategy could be incorporated as a 'prototype' or menu structure in a sophisticated CBT authoring sytem". Similarly, as others have observed (Van Pattern et al., 1986), the application of instructional strategies to computer-based delivery systems will require more work on creating prescriptions for developing non linear events such as "interactivity" and "on-line help".

IV. AUTHORING SYSTEMS

29. The efficient production of high-quality CBL material (couresware) has lagged far behind the development of hardware technology in computers. A survey by the National Centre for Educational Statistics (reported by Avner, et al., 1984) showed that the Apple II computer was the most-used computer in US classrooms. Despite the availability of computer-based instruction (CBI) languages for the Apple II, the survey noted that most of the courseware being used had been written in BASIC. Even though BASIC provided reasonable access to the computing capabilities of the micro, it was not designed for CBI use and meant that teachers not thoroughly familiar with the language could neither make full use of the computer or BASIC. The suggested solution was to augment BASIC with a set of commands which would add the CBI capabilities lacking in the language. In other words, develop some set of "authoring" commands which could address instructional strategy issues such as modes of student responding.

Authoring Languages

30. An authoring language is a programming language that includes specific commands for use in CBL. In PILOT these commands include "t:" (display this text); "a:" (accept answers); and "m:" (match to these answers), etc. In the central processor of the computer instructions are received in a very primitive, low level language called *machine* language. Because it is so difficult and laborious for people to write extensive programmes in machine language *higher-level* languages have been developed. These languages use instructions written in English to create the commands. For example, "READ" might mean to read information from a specified device such as a floppy disk drive; "DISPLAY" may mean to write what follows on the screen, and so on and so forth. In summary, a programming language is a collection of rules (the syntax to be followed) and words and symbols (the vocabulary) that allows the writer to give very precise instructions to a computer in a format that can be understood. The sequence of instruction is the *programme*. Although they are most often referred to as high-level languages, probably a more descriptive title would be "general purpose" languages, since they are designed to solve many classes of problems. Although there are literally hundreds of high-level languages, in education and training the most commonly used ones are BASIC and PASCAL.

31. All programming languages have their advantages and disadvantages. For example, because BASIC is an interpreted (not compiled) language (that is, it is executed by repeatedly scanning and 'interpreting' each character of the program text, rather than by being converted once into machine instructions), it is much more successful for incomplete, undebugged programmes, but very much slower than compiled programmes. However, the problem with most of them is that because they are general purpose languages (and in some cases structured around particular features of hardware systems, eg., FORTRAN and the IBM 704 computer), they cannot efficiently address educational needs such as judging the quality of a student's response. An alternative to using general-purpose languages is to design a language that can be used to write educational courses, or "courseware".

Authoring Systems

32. Authoring systems go a number of steps beyond authoring languages in the capabilities they provide. Many of them like TICCIT or WISE are interactive systems which enable the ISD author, with a minimum or no knowledge of computer programming to create a multi-media computer-based lesson. Unlike "languages", authoring "systems" offer the courseware developer additional facilities for creating graphics, animations, video and audio interactions, tests and manager systems. The idea behind the early authoring systems was the desire to separate course content from both computer programming and teaching strategies. In TICCIT the teacher was only concerned with the content while a teaching strategy was embedded in the software - in this case based upon a rule-example-practice sequence which was believed to be an appropriate instructional strategy for concept learning. Thus, the lesson author was releaved of the need to specify the detailed sequential steps for learning the concept by the provision of a "learner control

keyboard" in which the the pupil was supposed to decide for him or herself which sequence to follow. However, since many students are not very capable of deciding their own strategies in the absence of strategy training (Hunt, 1977), TICCIT had to try to 'understand' what the pupil was doing and have some notion of 'approved' sequences (Self, 1985). Ideally, more power and flexibility needed to be built into the system if it were to support the different learning needs of students and different instructional strategy expectations of courseware developers.

Measures of Authoring Productivity

33. Time estimates for developing CBT courseware are notoriously difficult to make. The most widely quoted estimate is that it takes between 200 to 500 work-hours (including instructional design and post-production evaluation phases) to produce one hour (that is, averaged across a group of students studying in a CBL mode) of moderately sophisticated training interaction. Of course this estimate will vary considerably depending on the experience of the courseware development team and the degree of sophistication available in authoring tools and software. Three dimensions are critical in assessing any CBT's productivity potential: (a) the power of the computing system; (b) ease of use of the authoring tools; and (c) the overall output that can be obtained from authoring effort.

34. **Power**. There are two considerations in this dimension. Firstly, there is the cability of the hardware system itself. Many PC -based authoring sytems (personal microcomputers), because of their lack of memory and storage facilities allow only one person to develop courseware at a time. In comparison, supermicros with 3 megabytes of memory and over 80 megabytes of storage are powerful enough to support 8 to 16 authors developing CBL courseware at the same time. These systems are obviously much more productive than systems which are restricted to a single person. Secondly, power can be measured in the degree to which new operations or subprograms can be added to the operating system (Zinn, 1974). For instance, some authoring systems allow the student to access programs and data bases outside the lesson they are currently working on. Not only does this increase instructional productivity by giving access to additional resources beyond those constructed by the courseware developers, but also enhances the potential for creating "real world" learning experiences.

35. **Ease of Use**. This is usually meant to mean how fast an authoring language or system can be mastered by coursware developers (Fairweather & O'Neal, 1984; Hillelsohn, 1984). Zinn (1974) has qualified the dimension by including "ease of entry", "ease of editing", and "ease of testing". Hunt (1985) goes further and adds, "ease of understanding authoring aids such as manuals and on-line helps".

36. Ease of entry generally refers to on-line, interactive entry. However, some advance preparation of material (in addition to that included in instructional strategy preparation)may be necessary for some systems which require a particular format sequence. The range of entry procedures include codified systems on the one hand (eg., PILOT) to menu and prompt driven sytems on the other (eg. WISE). Ease of editing suggests two areas of concern. One is the convenience of access to the original source file, and the other, the simplicity of method for making alterations. Being able to easily check a program by executing an edit at any point in the lesson, jumping about from one part to the next and being able to quickly duplicate any procedure, is a powerful contributer to course writer productivity. The third aspect of ease of use is a lot more subjective in its measurement and might be termed the "frustration" level. This element is determined by factors such as how well the logic of the authoring language/system meshes with the understanding of the courseware developer, and the internal consistency of the program (eg., always using the RETURN key as the input terminator, or always presenting prompts in the same order).

37. **Output**. Fairweather and O'Neal (1984) describe this dimension as the quantity and sophistication of the graphics and text displays, logical control and analysis structures, and the scoring and logging constructs which can be produced per work-hour of effort.

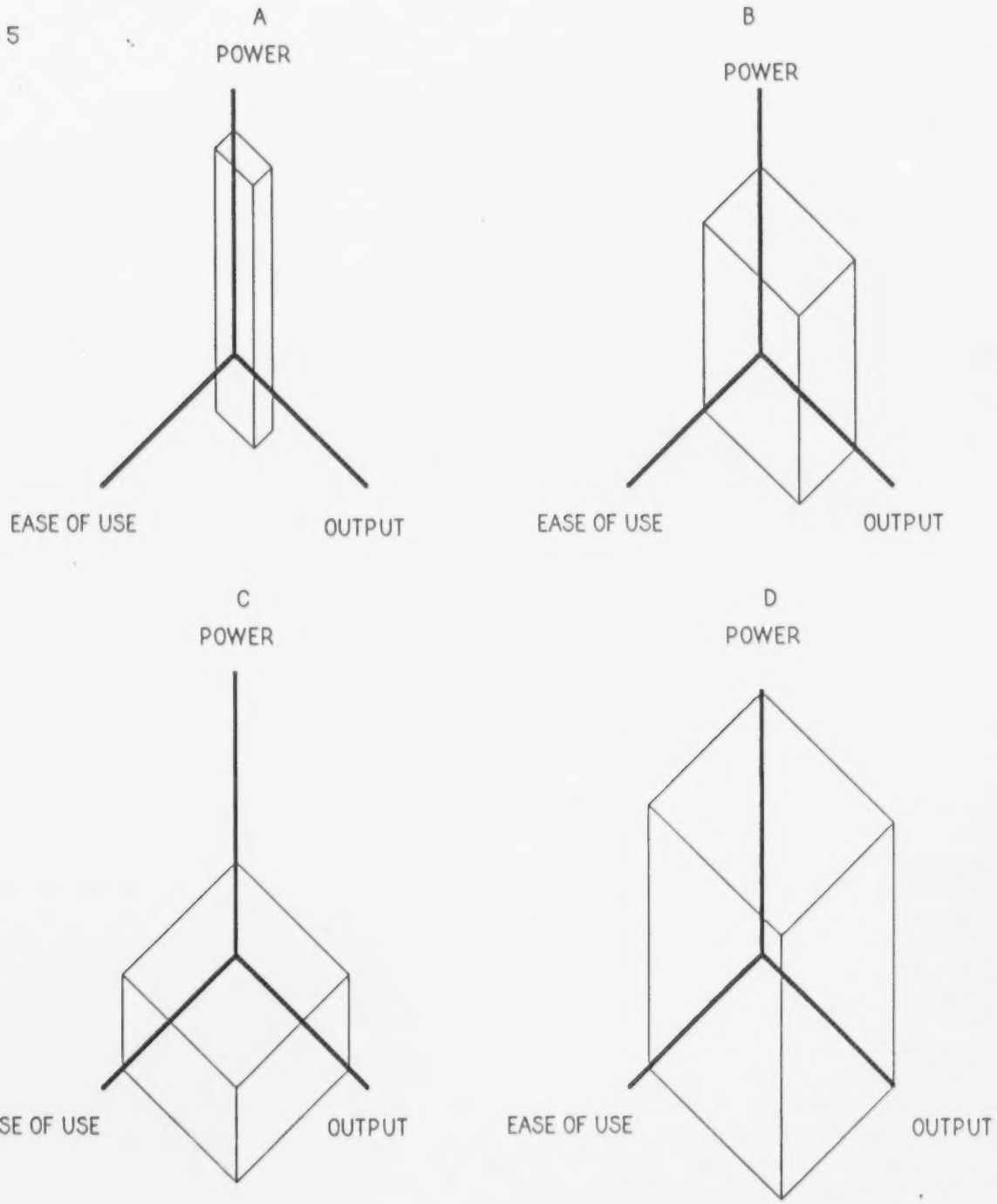
38. These authors (Fairweather and O'Neal, 1984) have conceptually demonstrated the relative contributions that different authoring systems can make to CBL productivity. Their model with minor modifications is illustrated in figure 5. In the first example (A) a general purpose language (like BASIC) will be relatively high in power. That is, using the language, a sophisticated programmer should be able to incorporate most of the design of the instructional strategy and utilize most of the capability of the computer system. However, "many of the functions would require relatively herculean effort, ingeneous programming solutions, and much re-inventing of special purpose wheels" (Fairweather & O'Neal, 1984, p.90). In fact, rarely have profesional programmers had access to sophisticated instructional design and instructional strategy prescriptions. As a consequence, although example (A) offers promise in principle, in reality courseware produced under these conditions has been technologically interesting, but instructionally ineffective.

39. In example (B) authoring languages (like SUPER PILOT) will provide the course developer with special commands for generating text and graphics displays but are typically restricted in the amount of computer capability they access. These languages have relatively easily keyed procedurees for accepting and judging student responses and thereby increase the overall usefulness and output which can be achieved. However, as Denenberg has said in relation to the TUTOR author language, "...while it is very easy for anyone to create a simple TUTOR program in very little time, it is extremely difficult for anyone to write a moderately complex program using the structure afforded by the TUTOR programming language...The more useful lessons require a degree of expertise not commonly found even in many professional programmers" (Denenberg, 1978).

40. Authoring systems included in example (C) such as PRIVATE TUTOR, tend to be PC-based programs, menu driven and well prompted and as a consequence easily learned and used. However, many of them are quite limited in the number and sophistication of displays and logical functions they support. PRIVATE TUTOR provides only four screen types, TEXT (with no interactive capacity), TRUE-FALSE, MATCHING, and FREE-RESPONSE. Writing to any of these screen types is a relatively simple matter of responding to prompts. Each of the screens allow for feedback to be given to student responses and branching for remediation or enrichment strategies.

41. The final example (D) incorporates all the advantages of the preceding three and provides for the incorporation of sophisticated instructional strategies which can be executed in a fraction of the time it would take if the routines needed to be individually programmed. TICCIT and WISE are the best examples of these hybrid authoring systems. Not only do these systems provide for online menu driven/prompt authoring, but also two additional facilities essential to high prouctivity. Firstly, a sophisiticated editing system which allows a production team to create and modify text and graphic material and insert additional resources such as audio or video inputs. Secondly, a 'course management function' (Self, 1985) which provides authors with ways of collecting data on student performance on the material. Systems like TICCIT recognize a universal lack of authoring experience and provide for instructional design guidance within the system itself. As a means of providing an interface between a coursewriter's level of expertise and an authoring system environment, TICCIT developed ADAPT. At the lowest level a novice coursewriter is introduced to a prompted environment that guides him or her in the selection of an appropriate instructional presentation mode and display. This is a highly prompted strategy. At the next level the same menus are used, but prompts are withdrawn. However, voluntary advise is available at any time through an advice option on the key board. The intermediate level generates menus of all available options and provides some complex sets of options, particularly for analyzing student responses. At the highest level, the coursewriter is assumed to be able to use the authoring language with its coding and syntax, while at the same time using menus for scoring and sequencing practice items.

Figure 5



Adapted from:
P.G. Fairweather and A.F. O'Neal
WICAT Systems, Utah.

An Authoring System Case Study: WISE.

42. The WISE authoring system is the fifth to have been developed by WICAT Systems since 1977. WICAT (standing for **W**orld **I**nstitute of **C**omputer **A**ssited **T**eaching) was formed not by computer engineers, but a group of educators, instructional designers and computer learning scientists, dedicated to finding out ways in which technology could improve education and training. Guided by Dr Dustin Heuston, a successful school principal for New York, who was convinced that traditional teaching and training methods could be improved, he and a team of visionaries set to work to see how the new technologies such as microcomputers and videodiscs might contribute to the development of more effective learning. One of their early successes came in 1979 with the award of a US Defence Advanced Research Projects Agency contract to develop an advanced technology training system. The outcome of this work proved that the system they had created could support the kinds of education and training products they wished to produce.

43. Like TICCIT's ADAPT and TAL (TICCIT's Advanced Language), WISE (standing for WICAT's Interactive System of Education) is a menu driven, prompted and helped system which is easy to learn and allows for a high degree of coursewriter productivity. Sophisticated instructional strategies involving complex displays and logic can be implemented easily and quickly. Further, as well as the range of options available in the menu including a wide variety of programming utilities and tools, WISE will allow the courseware developer to interactively access other data bases or use other general languages such as BASIC, PASCAL or C, outside its own system.

44. **Power**. WISE is an extremely powerful system which enables the author to perform literally any function of which the computer system for courseware development is capable. However, in that the computer must be capable of addressing text and graphics material, audio, videodisc and an array of other utilities, standard PC systems have neither the computing power or storage facilities to address the drivers embedded within the WISE system. Consequently, WICAT built its own series of supermicro (mini-frame) computers based on an MC68000 32/16 bit processor. These systems, ranging from the S1250 up to the S1260 for computer-based training applications, provide for between 1 to 7 megabytes of memory, and from 48 to 516 megabytes of data storage. This power contrasts with the typical microcomputer which has between 0.128 and 0.512 megabytes of memory and 0.360 to 20 megabytes of storage! An individual piece of courseware such as English as a Second Language (ESL) in its early versions may take up to 40 megabytes of storage. Add 12,000 graphics from a reading program and the storage requirement jumps to 76 megabytes. Multiple courseware systems which are typical packages in WICAT's computer-based education programme mean that the minimum hardware configuration needed to run the courses is not less than 160 megabytes of storage and 3 megabytes of memory (Heuston, 1985). In an authoring configuration for CBT, even the smallest system (the S1250 with 3 megabytes of memory and an 86 megabyte hard disk) will allow 8 or 16 authors to work on one course, or to develop 8 to 16 courses simultaneously. This is a very significant consideration for CBT productivity. PC-based systems simply don't have the power to produce this level of authoring impact on training.

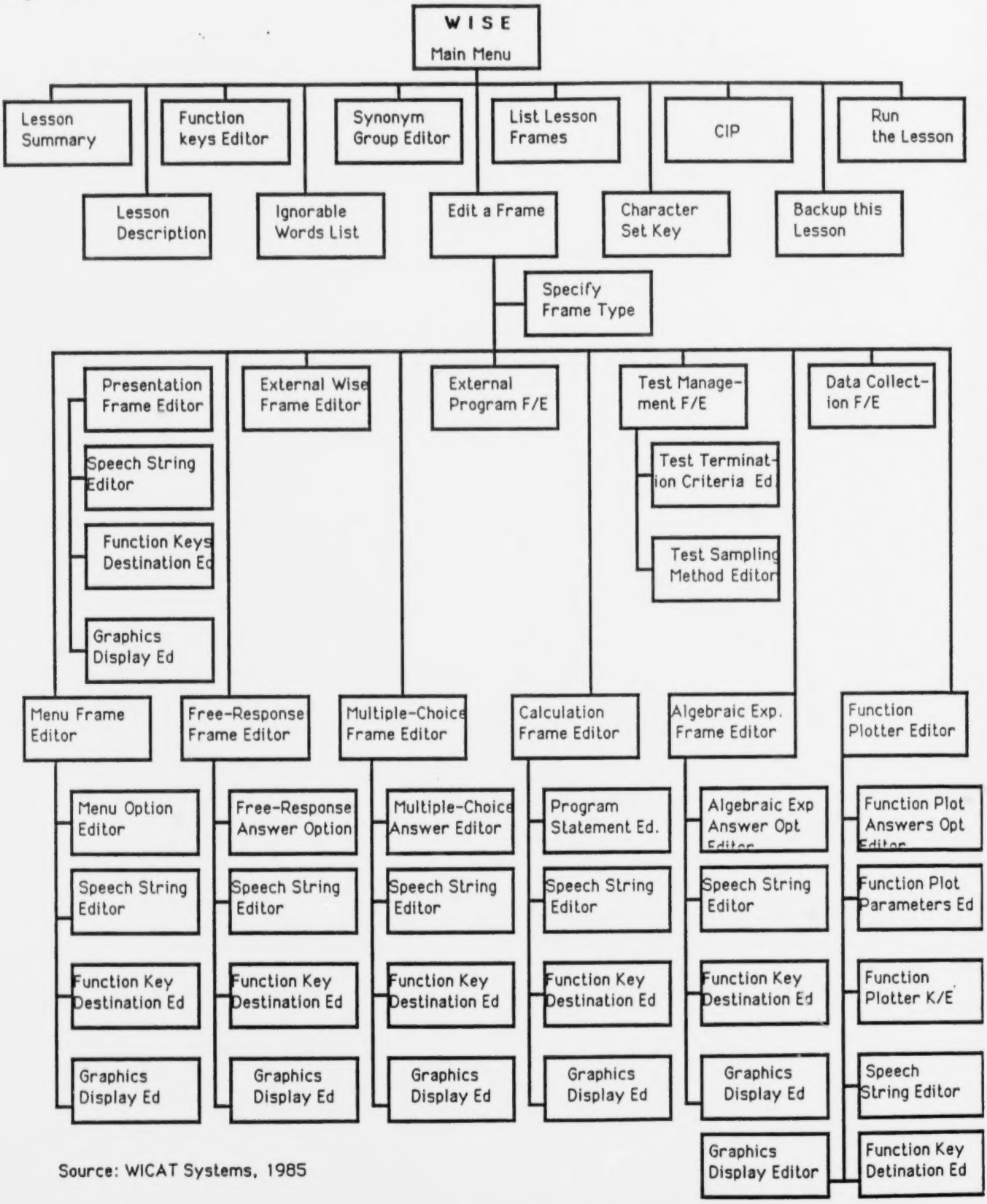
45. The basic unit of instruction in WISE is called a *frame*. A frame may be accessed and developed at any of three different levels of authoring complexity:

46. **Level One** - this is a highly prompted set of menus. No computer coding is required. Many CBL authors using the system never go above this level in developing their CBT lessons.

47. **Level Two** - this level involves the use of the *calculation frame*. Within this frame the author has access to an authoring syntax which allows computations, creation of animated displays,

Figure 6

THE *WISE* AUTHORING SYSTEM



Source: WICAT Systems, 1985

conditional branching, use of system variables, acceptance of open-ended student responses, answers judging the quality of students' responses, calling the logic and and display of other lesson frames as subroutines (nested at any level), control of external devicessuch as videodisc players, manipulation of common variable pools accessible to any set of users (simultaneously), and many other functions. The calculation frame syntax is a powerful "authoring language" in its own right, and just as entire lessons can be written without access to any 'calc frame', most lessons could also be written using only calc frames, if so desired (Fairweather & O'Neal, 1984).

48. **Level Three** - any lesson frame may be designated as an external frame. When the student is branched into an external frame, WISE will execute any program or operating system neamed in the frame. This may range from using a program based on an operating system like PICK, to languages such as BASIC, PASCAL or C.

49. **Ease of Use** - in terms of "ease of entry" WISE facilitates the process by providing two entry points. First, the display screen contains all the instructional material visible to the learner. The author creates all text, graphics and animations through simple two-key stroke commands. For example, the key strokes "CT" (create text) tells WISE that the following input is text material to be displayed on the screen. These key stroke commands can be further simplified by using a single key-stroke on the alternate key pad. In this mode the key "8" can replace the two key strokes C and T. Graphics can be created by using two-key commands or alternative single function keys. For example, to draw a box (using an IBM PC keyboard) the author would press the "-" key to create the box graphic, and then manipulate this image in terms of size and shape by using the arrow keys. To animate an object after it has been drawn on the screen, the author uses three commands; "user create" (UC); "event create" (EC); and "path create" (PC). For each of these commands WISE will prompt the author for information such as "do you want to rotate this object?" "What path do you want it to take on the screen?" "Do you want it to leave a trail as it moves about the screen?" Using these simple keys it is possible to create quite complex animations in only a matter of a minute or two.

50. The second point of entry is to structure the logic of the lesson, that is, to structure the sequencing of frames, the judging and scoring of student responses, and selecting utilties to be incorporated within the lesson like videodisc and speech sequences. Each display frame has a number of logic menus which allow the author to tell WISE how he or she wants the particular frame to fit into the lesson as a whole. For example, option "6" on the "Presentation Editor Menu" will prompt the author for the "name of the next frame". The author will respond by typing in the name of the next frame that the student will go to (figure 7).

51. At any stage of courseware development the author can run the program in a "debug" mode. This is the manner in which the student would view the lesson, except that each frame name appears at the bottom of the screen so that the author can identify those which need further editing. If the lesson contains a bug which would prevent the lesson from running properly, WISE will terminate the lesson and give an error message describing the problem and identifying the name of the frame where the problem has occured. A typical message might read as follows: *An error has occured in WISE 2.2.2. It has occured at station #6. The error is in the lesson named 'DAFT' at' frame #35. A -1 was the time limit for a menu frame. The string RESPONS is to be used as the menu response. But the string does not match a menu option. Please report these numbers. ER=423 SN=21001004 MG=240. The process was terminated with an error.* (The reference number alerts WISE Customer Support Service to identify the specifics of the problem should the author not be able to fix it his or herself.) Editing a display frame is further simplified by the facility of WISE to label every object (graphics and text) that appears of the screen. All that is required is for the author to type in a command (eg., "OD" - object delete) and WISE will ask for the name of the object (its identifying letter) to be modified.

52. **Output**. WISE's philosophy is that once a procedure has been created it should never have

to be created again. For example, when an author makes a graphic which she will want to use again, that graphic can be stored in a graphics library and accessed at any stage of the lesson development, or at any stage in the development of any other lesson. Whole display frames and or frame logic can be copied from within the current lesson, or from another lesson.

53. Some frames such as the "Free Response" may require a huge range of student responses to be judged. For each category of answer, an answer option file as shown figure 7 has to be completed by the author. Much of the information will be common to all answers, such as the significance of word order (18) and spelling/value tolerance (10). These criteria provided by the courseware developers can be templated once and copied for use time and time again. Each answer option can then be quickly tailored for the unique responses and feedback that individual learning strategies may require.

54.

Although all the courseware can be developed in non-coded structures, some authors with sophisticated programming skills may want to use calculation frames to write original sequences. Once written, these can be made available to other non-programming users via the normal menu and prompt system. This facility greatly enhances the quality and range of output accessible to all authors.

Figure 7 WISE's ANSWER OPTION EDITOR

```
Answer Option Editor          #5          FREE
Curr. Answer Opt.: UNEX                   Total Answer Opt.: 0

1) Ques. Desc.  1        3) Temp. Opt.  NONE        5) Max. Dest.
2) Opt. Desc.   0        4) Max. Att.   0           6) Resp. Type  STRING
9) Scoring Weight 0                    10) Spell/Val Tolerance    100
7) Answer to be matched                11) Judgement if matched  UNEXPECTED
                                       12) Next Frame if Matched  FREE
                                       13) Space Character Sig.?        NO
                                       14) Synonym/Sci Notation         NO
                                       15) Use Ignorable Word List      NO
8) Feedback if matched                 16) Punctuation Sig.? Min Sig Dig NO
                                       17) Capitalisating Sig?Min Sig Dig NO
                                       18) Word Orders Sig?             YES
                                       19) Extra Words Allowed?         NO
                                       24) Super/Sub. Ord Sig?          NO

20) ADD  21) INSERT  22) DELETE  23) COPY

Enter Choice or Press <ESC> To Quit
```

55. Systems such as WISE offer instructional designers and training resource managers not only the possibility of designing and delivering sophisticated learning environments (Hunt & Hunt, 1986), but also the management of information such as student performance, lesson data, score averages and the like, which contribute to the rigorous monitoring of student learning progress. This latter "computer-managed instruction" (CMI) feature makes the product not only an effective method for delivering training, but also for mananging the human resource development consequences of CBT interventions.

56. Although WISE courseware development is authored via powerful minicomputers such as the WICAT S1255, the lessons can be disseminated either through terminals linked to the mini-computer, or through standalone PC microcomputers. This means that lessons can either be used in formal classroom training environemnts, or in less formal, dispersed and even "distance" learning situations. In the PC mode, lessons are downloaded onto floppy disketts via an emulator program. With the most recent versions of PC WISE, drivers can address a wide range of enhanced graphics PC systems, giving 16 colour (including video) output. For the price of a PC microcomputer students can access a level of program sophistication which only a few years ago was beyond even the most expensive main-frame systems.

V. THE ADVANTAGES OF CBT

57. As we stated at the beginning of this paper, we are at the start of a new era in education and training. The traditional delivery system of teachers and books has developed about as far as it can go. "In the United States, between 1950 and 1975 the percentage of the gross national product spent on education rose from 3.4 to 7.4 percent, an increase of 118 percent. During this period, measurable benefits from this investment appear to be negligible. Generally speaking, student scores went down. In other words, having more than doubled the expenditure for the educational delivery system there was no improvement" (Heuston, 1985, p.7.). Computer-based learning systems offers the promise of delivering greater levels of learning achievment, more professional teacher involvement and higher pupil-parent satisfaction from education and training than at any previous time. Greg Kearsley (1983) has identified the benefits which are likely to flow from CBT implementation as eleven measurable outcomes.

58. **Increased Control**. Many organizations expend considerable sums of money in developing courses to train staff in new procedures and equipment. However, manpower turn-over often exceeds the speed and cost that would be incurred if the course was to run each time new staff were appointed. Alternatively, trainers, receiving the material modify it in the light of their own foibles, sometimes improving it, but often, over a period of time reducing its instructional effectiveness. With CBT the cost of mounting a training programme is negliable (after the initial development costs), and with standalone delivery systems (PC's) the need for expensive outlays associated with formal training (travel, accommodation, student perdiems, etc.) is significantly reduced. Further, the quality invested in the initial development is not going to be dissipated by random modification or cost-saving reductions to its content. Quality control is enhanced through the computer's ability to objectively score and monitor student progress and highlight deficiencies in instructional strategies.

59. **Reduced Resource Requirements**. CBT can generally achieve learning mastery of skills in a much shorter duration than conventional methods. Furthermore, with systems which allow the dissemination of content on inexpensive floppy disks, there is no limit to the number of students who can be engaged in the learning process, in the absence of highly paid instructors. In some cases, where the cost of equipment may outprice the cost of training staff, CBT programs

with sophisticated simulation procedures may make it acceptable to forgo the need to procure expensive equipment for teaching purposes. More money can thereby be dedicated to upgrading teacher resources.

60. **Individualization**. One of the accepted determinants of successful learning is the degree to which each student is able to work to his or her own pace and receive the kind of feedback most useful for achievement. Heuston (1985) has reported that depending on the length of the school day, and the traditions of an individual school, only about 28 percent of the total delivery time is spent in instruction (or about 92 minutes a day), and of that time, at best 60 to 120 seconds may be individualized for a student. CBT allows for 100 percent individualized instruction, and with some systems, learner-controlled instruction.

61. **Timeliness and Availability**. A common problem in training is that it often cannot be provided when and where it is needed. Because of the lack of qualified trainers, equipment, teaching accommodation and other resources, trainees are forced to wait. Sometimes, when it is essential that the trainee takes up the job because of overall manpower considerations, in the absence of appropriate training provisions, either poor on-the-job teaching is offered , or no training is provided at all. Clearly this is an area in which CBT has a tremendous potential for providing training "on demand".

62. **Reduced Training Time**. The capability of CBT to reduce training time is one of its most frequently published attributes. Studies reported by Kearsley (Orlansky and String, 1979) demonstrate that while the actual range of savings may vary considerably, the median value is about 30 percent. It is obvious that if savings in time can be made of this order, considerable savings are likely to accrue in increased trainee productivity, reduced trainer salaries, and reduced needs for large and costly training infrastructures.

63. **Improved Job Performance**. Because of the interactive and individualized nature of CBT, its inherent potential to improve the quality of training is significant. As earlier research has shown (Hunt, 1984), traditional approaches to teaching often result in very little actual learning taking place. This is partly because under conventional conditions students can avoid attending to the instruction for large periods of time. But in interactive individualized instruction the learner is forced to engage in the process by the very nature of the delivery system.

64. **Convenience**. Computers are becomming increasingly common in every day work places. As a consequence, it is becomming more and more possible to "embed" the CBT program in the work system. For example, project managers who are required to supervise accounting procedures may have an instructional package on accounting procedures loaded into the accounting program.

65. **Change Agent**. Computers have introduced a new language into the world. It is said that young people need to be computer literate today if they are going to have much chance of succeeding in a technological environment. CBT is part of the process of providing that literacy, implicitly across all its content areas, and explicitly through the use of specific courseware like WICAT's "Introduction to Computers" program.

66. **Increased Learning Satisfaction**. Because of the non-threatening, interactive and achievement orientated nature of CBT, students typically find it more motivating than other forms of instruction. One of the major reasons for student disenchantment with learning is previous, and often, continuous failure. Similarly, nothing breeds successes better than success itself. If students can be placed in non-threatening, interactive environments, with tutors (the computer) who never get bored with them, nor impatient, no matter how many times they make a mistake, the chances are they will want to learn more.

67. **Reduced Development Time.** This last advantage is now only just beginning to be

realized with the development of hybrid authoring sytems like TICCIT and WISE. As technology change increases, so too will the need to constantly revise and update even traditionally delivered training programmes. Until recently, revising material in CBT programs was an extremely costly process. However, with the editing facilties previously described the revision process can be quick and deletions and insertions of content streamlined. Coupled with the feedback that embedded test facilties can provide, not only can content be changed to meet changed circumstances, but also the quality of the instruction itself to meet increased understanding of the human information-processing system.

VI. CONCLUSIONS

68. In this paper we have focused on the design of prescriptive strategies for developing CBT courseware. Throughout the paper we have said very little about the teacher other than to observe that by and large their role characteristics have not changed much over the past 500 years. In the early days of CAI (computer-assited instruction) it was thought that computer-based learning and its attendant philosophy of individualized learning would be so irresisitible that, as Bunderson put it, it would "alter the culture of the school" (Bunderson, 1981, p.95). It never happened. In fact, only in those circumstances where teachers have been prepared to accept the computer as part of a teacher-centred classroom culture, has the revolution flourished (Wilkinson & Patterson, 1983). The WICAT experience with its laboratory "Waterford" school has been very similar. There, the computer is used in a regular curriculum programme for only about one-hour per day per student. Teachers who have been hired for their pedagogic ability, rather than any particular interest with computers, are trained in the application features of the sytem; adaptive testing; student management; curriculum development; etc., rather than anything about the computer itself. Computer studies is left to the teachers who are expert in computer programming!

69. But the revolution will change teachers, at every level of the system. It will never replace them. Teachers may not need to be expert in computer programming, but increasingly they will need to become aware of (and for those who wish to develop courseware) expert in, the science of instructional design and its development. And for those who accept this incalculable resource in their classrooms, workshops and laboratories, their teaching role is likely to be much more dynamic, much more professional and much more personally satisfying than at any time in the history of the profession. Even more excitingly, they will finally have at their disposal the means for ensuring that every individual achieves personally, socially and economically, in ways never before thought possible.

REFERENCES

Ashcroft, E.R. (1983) **Creating Learning: A Handbook for Instructor Trainers.** Singapore: ADB-BERL.

Avner, A., Smith, S., and Tenczar, P. (1984). CBI authoring tools: Effect on productivity and quality. **Journal of Computer-Based Instruction**, 11, 3, 85-89.

Block, J.H. (1971). **Mastery Learning: Theory and practice** New York: Rinehart and Winston.

Botkin, J.W., Elmandrja, M., and Malitza, M. (1979). **No Limits to Learning: Bridging the Human Gap.** Oxford: Pergamon Press.

Bloom, B.S. (1976). **Human Characteristics and School Learning**. London: Cambridge University Press.

Bunderson, V.C. (1980). Instructional strategies for videodisc courseware: One McGraw-Hill disc. **Journal of Educational Technology Systems**, 8, 207-210.

Bunderson, V.C. (1981). Courseware. In H.F. Oneil, Jr. (Ed.), **Computer-Based Instruction: A State of the Art Assessment**. New Yor; Academic Press.

Carkhuff, R.R. (1983). **Sources of Human Productivity**. Amherst: Human Resource Development Press.

Cronbach, L.J. and Snow, R.E. (1977). **Aptitudes and Instructional Methods: A Handbook for Resarch on Interactions**. New York: Irvington.

Crowder, N.A. (1962). The rationale of intrinsic programming. **Programmed Instruction**, 1, 3-6.

Denenberg, S. (1978). A personal evaluation of the PLATO system. **Sigcue Bulletin**, 12, 2, 3-10.

Fairweather, P.G., and O'Neal, A.F. (1984). The impact of advanced authoring systems on CAI productivity. **Journal of Computer-Based Instruction**, 11, 3, 90-94.

Heuston, D.H. (1985). **Some Critical Issues in Introducing Computer Technology into Schools**. Provo, Utah: WICAT Systems.

Hillelsohn, J. (1984). Bench marking authoring systems. **Journal of Computer-Based Instruction**, 11, 3, 95-97.

Hunt, G.J.F. (1977). The effect of instructional preference on student attitudes. **New Zealand Journal of Educational Studies**, 12, 164-175.

Hunt, G.J.F. (1984). **Assessing Needs in Vocational Teacher Training**. Singapore: ADB-BERL.

Hunt, G.J.F. (1984b). NEBEAT - the systems appraoch updated. **PLET-Programmed Learning and Educational Technology**, 21, 1, 53-60.

Hunt, G.J.F. (1986a). Identification of trainer competencies for access programme training of trainers: A survey of supervisors on Work-Based Programmes. **Occasional Paper**. Palmerston North: Massey University.

Hunt, G.J.F. (1986b). Needs assessment in adult education: tactical and strategic considerations. **Instructional Science**, 15: 287-296.

Hunt, L.M. (1975). Authoring Languages and Systems. **Occasional Paper**. Palmerston North: Massey University.

Hunt, G.J.F., and Hunt, L.M. (1986). Computer-based training. **Interface**, December, 50-51.

Kearsley, G. (1983). **Computer-Based Training. A Guide to Selection and Implementation**. Reading, Massachusetts: Addison-Wesley Publishing Company.

Lumsdaine, A.A., and Glaser, R. (1960). **Teaching Machines and Programmed Learning; A Source Book**. Washington, D.C.: National Education Association.

Mager, R.F. (1962) **Preparing Objectives for Programmed Instruction**. San Francisco: Fearon.

Merril, M.D. (1980). Learner control in computer-based learning. **Computers and Education**, 4, 77-95.

Monjan, S.V., and Gassner, S.M. (1979). **Critical Issues in Competency Based Education**. New York: Pergamon Press.

O'Neal, A.F. (undated). **The Current Status of Instructional Design Theories in Relation to Today's Authoring Systems**. Orem, Utaha: WICAT Systems.

Reigeluth, C.M. (1979). TICCIT to the future: Advances in instructional theory for CAI. **Journal of Computer-Based Instruction**, 6, 2, 40-46.

Reigeluth, C.M. (1983). The elaboration theory of instruction. In C.M. Reigeluth (Ed.). **Instructional-Design Theories and Models: An Overview of their Current Status**. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Rogers, C.R. (1968). **Freedom to Learn**. Columbus, Ohio: Merrill.

Self, John. (1985). **Microcomputers in Education: A Critical Appraisal of Educational Software**. Brighton: Harvester Press.

Van Pattern, J., Chao, Chun-I., and Reigeluth, C.M. (1986). A review of strategies for sequencing and synthesizing instruction. **Review of Educational Research**, 56, 4, 437-471.

Whitehead, A.N. (1932). **The Aims of Education** (Paperback edition, 1962). London: Benn.

Wilkinson, A.C., and Patterson, J. (1983). Issues at the interface of theory and practice. In A. C. Wilkinson (Ed.), **Classroom Computers and Cognitive Science**. New York: Academic Press.

Zinn, K.L. (1974). Requirements for effective authoring systems and assistance. **International Journal of Man-Machine Studies**, 6, 381-400.